REMARKS

Claims 1-4, 6-11, and 14-19 are all the claims pending in the application. By this Amendment, Applicant amends claims 1-4, 6-11, 14, and 16-19 to clarify the features set forth therein.

I.   Summary of the Office Action

The Examiner objected to claim 1 and rejected claims 1-4, 8-11, and 14-19 under 35 U.S.C. § 103(a).

II.   Claim Objection

Claim 1 is objected to because allegedly the use of "if" in the claim elements makes them conditionally executable statements, that may not be true when the condition is not met. Applicant has revised the claim (including 1-4, 6, 7-11, 14, and 16-19), and respectfully submits that the claims as now presented no longer include the potential informality mentioned by the Examiner. Applicant therefore respectfully requests the Examiner to withdraw the objections to the claims.

III.   Claim Rejections under 35 U.S.C. § 103(a)

The Examiner maintains the rejection of claims 1-4, 6-11, 18, and 19 under 35 U.S.C. § 103(a) as being unpatentable over Crow et al. (US 2002/0161915 A1), hereinafter referred to as "Crow" in view of Ganesan et al. (US 2003/0069973 A1), hereinafter referred to as "Ganesan", Varma et al. (US 2004/0037302 A1), hereinafter referred to as "Varma", and further in view of Rana et al. (US 2002/0095512 A1), hereinafter referred to as "Rana" and claim 17 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Crow in view of Ganesan, Varma, and Rana, and further in view of Hui et al. (US 2004/0151197 A1), hereinafter referred to as "Hui".

Applicant respectfully traverses these grounds of rejections at least in view of the following

exemplary comments.

Independent claim 1 *inter alia* recites:

> A method for receiving a plurality of packets
> from a network and distributing the packets to a
> plurality of protocol processors comprising the
> steps of:
>
> when a received packet is a fragmented packet,
> determining whether the received packet is a
> first fragment packet;
>
> when the received packet is the first fragment
> packet, looking-up a tunnel ID of the received
> packet and a fragment IP header of the received
> packet, and comparing the result of the looked-up
> fragment IP header with each list of a fragment
> look-up table into which the results of fragment
> looked-ups for other received packets are
> entered, to determine if there is a corresponding
> list;
>
> searching an index indicating one of the protocol
> processors and corresponding to the tunnel ID of
> the received packet from a tunnel ID look-up
> table, and when the list corresponding to the
> result of the looked-up fragment IP header exists
> in the fragment look-up table, entering the index
> into the corresponding list of the fragment look-
> up table;
>
> wherein when the list of a fragment look-up table
> is not accessed during a predetermined time, the
> list is removed from a fragment look-up table,
>
> wherein when the received packet is not the first
> fragment, looking-up the fragment IP header of
> the received packet and comparing the result of
> the looked-up fragment IP header with each list
> of the fragment look-up table, to determine if
> there is a corresponding list; and
>
> wherein when the list corresponding to the result
> of the looked-up fragment IP header exists in the
> fragment look-up table, performing operations of:
>
> determining whether the index corresponding to
> the result of the tunnel ID look-up exists in the
> corresponding list; and

```
attaching the index as a tag to the received
packet; and transmitting the received packet to
the corresponding one of the protocol processors,
when the index exists in the corresponding list.
```

The Examiner acknowledges that Crow in view of Ganeson do not disclose or suggest the above-emphasized features of claim 1. The Examiner, however, alleges that Varma cures these deficiencies. Specifically, the Examiner alleges that Varma describes in paragraph [0032] "a method wherein if the received packet is the first fragment packet, looking-up a fragment ID of the received packet, and comparing the result of the looked-up fragment ID with each list of a fragment look-up table into which the results of fragment looked-ups for other received packets are entered, to determine if there is a corresponding list; and if the received packet is not the first fragment, looking-up the fragment ID of the received packet, and comparing the result of the looked-up fragment ID with each list in the fragment look-up table, to determine if there is a corresponding list." (*see* pages 8-9 of the Office Action). Applicant respectfully disagrees.

Varma describes a process associated with an en-queuing operation (adding data). Once the address of the queue is received (300), the control data (head, tail and count) for the associated queue is received from the control memory (310). A block in data memory from the free list is allocated to store the data (320). A determination is made as to whether this is a first block of data associated with the queue (330), that is, if the queue was empty upon the arrival of the data. This is achieved by checking the count value for zero (0). If the count value is zero (330 Yes), the head address and the tail address are both set to the address of the allocated block (340), as this is the only data item (packet) associated with the queue. If the count value is non-zero (330 No), the link in the link memory corresponding to the current tail of the queue is modified to point to the newly allocated block in data memory (350) and the tail address is set to the allocated block (360). The data is then added to the block (370) and the count value is

incremented (380). The updated control data (head, tail and count) is then stored in the control

memory (390) (Fig. 4; ¶ 32).

Varma only describes that the control memory stores control data of a packet (head, tail,

count), and, if a count value is 0, a block allocated to the data memory is determined to be a first

block of data associated with a received queue, and a head address and a tail address included in

the control memory is set to an address of the allocated block, and if the count value is not 0, a

link in the link memory corresponding to the current tail of the received queue is modified to

point to a newly allocated block in the data memory and the tail address is set to the address of

the allocated block. In other words, Varma describes checking the count value of the control

memory, which is considered as corresponding to the fragment lookup table by the Examiner,

determining whether the block allocated to the data memory is the first block of the received

queue or not according to whether the count value is 0 or not, if the count value is 0, setting the

head address and the tail address included in the control memory to the address of the allocated

block, and if the count value is not 0, modifying the link in the link memory corresponding to the

current tail of the received queue to point to the newly allocated block in the data memory and

setting the tail address to the address of the allocated block. Also, Varma, describes that other

fields of the control memory include a count of the packet but do <u>not</u> include <u>a message ID</u>.

Accordingly, Varma fails to disclose looking up <u>a fragment IP header</u>, and <u>comparing the</u>

<u>result of the looked-up fragment IP header with a fragment lookup table</u> into which the results of

fragment looked-up for other received packets are entered, to determine if there is a

corresponding list. That is, Varma does not disclose or suggest determining whether a received

packet is a first fragment packet or not, looking up a fragment IP header of the received packet,

and comparing a list resulting from the looked up fragment IP header with a list of the fragment

lookup table of other packets, to determine if there is a corresponding list. Varma only discloses

checking whether <u>a count value</u> included in the control memory and does not disclose or suggest

operations with the <u>fragment IP header</u> as set forth in claim 1. In short, Varma does not cure the

above-identified deficiencies of Crow and Ganesan. Rana is only cited for its alleged description

of entering an index (*see* pages 11-12 of the Office Action) and Hui for its alleged description of

setting the index to invalid (*see* page 44 of the Office Action) and as such these references do not

cure the above-identified deficiencies of Crow, Ganesan, and Varma.

In view of the above, Crow, Ganesan, Varma, Rana, and Hui fail to disclose or suggest

each and every feature of claim 1. Therefore, claim 1 is patentable over Crow in view of

Ganesan, Varma, Rana, and Hui. Claims 2-4 and 6 are patentable at least by virtue of their

dependency on claim 1.

Independent claims 7 and 14 recites features similar to those recited in claim 1.

Therefore, claims 7 and 14 should be patentable for at least analogous reasons. Claims 8-11 and

15-19 are patentable at least by virtue of their dependency on claim 7 and 14, respectively.

IV.   Conclusion

In view of the above, reconsideration and allowance of this application are now believed

to be in order, and such actions are hereby solicited. If any points remain in issue which the

Examiner feels may be best resolved through a personal or telephone interview, the Examiner is

kindly invited to contact the undersigned attorney at the telephone number listed below.

The USPTO is directed and authorized to charge all required fees, except for the Issue

Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any

overpayments to said Deposit Account.

Respectfully submitted,

SUGHRUE MION, PLLC   Nataliya Dvorson
Telephone: (202) 293-7060   Registration No. 56,616
Facsimile: (202) 293-7860

WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Date: **July 28, 2010**